

Práctica 1 Aprendizaje Automático

Antonio Álvarez Caballero

Generación y visualización de datos

En primer lugar, creamos una función que crea un data.frame con valores aleatorios según una distribución uniforme. N es el número de filas del data.frame, dim el número de columnas y $rango$ el rango donde estarán los valores.

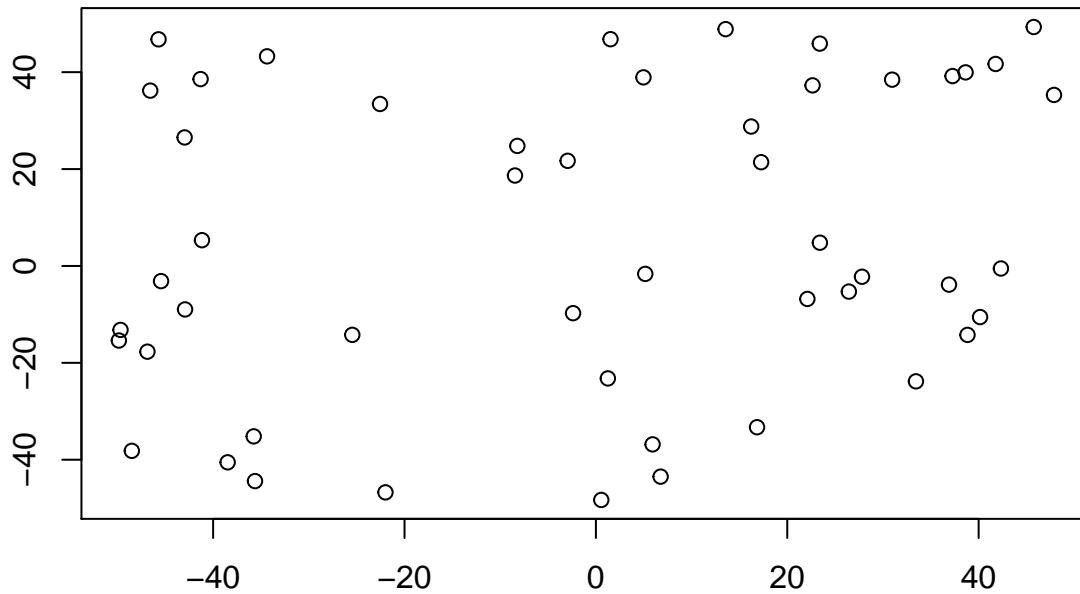
```
simula_unif <- function(N, dim, rango){  
  res <- data.frame(matrix(nrow = N, ncol = dim))  
  
  for(i in 1:N){  
    res[i,] <- runif(dim, rango[1], rango[length(rango)])  
  }  
  
  return(res)  
}
```

Del mismo modo creamos una función análoga para la distribución *normal* o *gaussiana*.

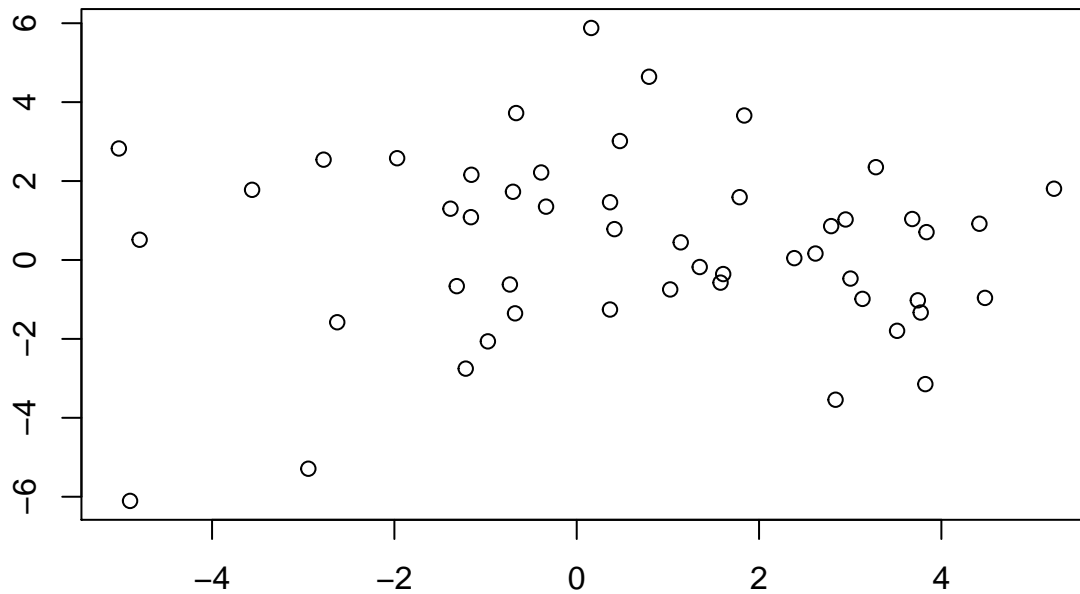
```
simula_gauss <- function(N, dim, sigma){  
  res <- data.frame(matrix(nrow = N, ncol = dim))  
  
  for(i in 1:N){  
    res[i,] <- rnorm(dim, sd = sqrt(sigma))  
  }  
  
  return(res)  
}
```

Ahora asignamos los resultados a sendas listas y las dibujamos.

Distribución uniforme



Distribución normal



Ahora escribimos una pequeña función para calcular una recta dados dos puntos. Daremos la recta con la ecuación punto pendiente, por lo que tenemos que calcular la pendiente y la desviación.

```
simulaRecta <- function(intervalo){  
  A <- runif(2, intervalo[1], intervalo[length(intervalo)])  
  B <- runif(2, intervalo[1], intervalo[length(intervalo)])  
  
  m <- (A[2] - B[2]) / (A[1] - B[1])  
  b <- A[2] - m * A[1]
```

```
  return(c(m,b))  
}
```

Generamos un par de números aleatorios y generamos una recta

```
coeff <- simulaRecta(-50:50)  
  
plot(lista1[,1], lista1[,2], main = "Holi", xlab = "", ylab = "")  
abline(a = coeff[1], b = coeff[2])
```

